USB Mobile-Friendly Device Design Guide

Revision 1.0

03/11/98

Revision History

Revision Number	Issue Date	Content/Changes
0.9	November 18, 1997	Initial review distribution version
0.91	December 3, 1997	Second review distribution version
0.92	December 24, 1997	Third review distribution version
0.93	January 14, 1998	Fourth review distribution version
1.0	March 11, 1998	Release 1.0 distribution

Mobile-friendly USB Devices

Introduction

Mobile platforms are inherently limited in the total power available to them. Therefore, a mobile platform is constantly balancing between two opposing goals: features and performance versus battery life. A mobile platform is not useful if its perceived performance does not satisfy the user, or it is not capable of performing time critical operations. Its usefulness is similarly limited if its battery life is too short for a user to feel he can accomplish meaningful work without being attached to a wall plug. As a new technology like USB becomes a standard feature on the mobile platform, it's important to consider the power consumption draw and its impact on battery life. Analysis has demonstrated that one USB device can consume more than 10% of a system's entire power budget, as a worst-case scenario. It is imperative that mobile-friendly design features are applied to developing USB devices in order to preserve as much power as possible, while also providing all functionality to satisfy the user's needs.

Scope

Certain characteristics will enhance a USB device's "friendliness" for a mobile user. This document identifies those characteristics that set a "mobile-friendly" device apart. It also describes the unique properties and behavior that device designers might expect from a mobile platform operating on limited battery power.

References

- "PC98 System Design Guide", http://www.microsoft.com/hwdev/hwdev4.htm
- "Universal Serial Bus Specification", Revision 1.0, http://usb.org/developers/
- "USB Mobile Design Guide", Revision 1.0, http://developer.intel.com/design/usb/designex/usbgl10.htm
- "OnNow Power Management and USB", http://www.microsoft.com/hwdev/pcfuture/

Definitions

- **USB** Universal Serial Bus is a 5.0V, serial I/O cable bus, with a hierarchical tree structure.
- **High-Powered** devices draw more than 0.5W and no more than 2.5W from the USB cable.
- **Low-Powered** devices draw no more than 0.5W from the USB cable.
- **Bus-Powered** devices draw all of their power from the USB cable. High-power devices may draw up to 2.5W maximum.
- **Self-powered** devices have a non-USB power source (for example power line or internal battery). Self-powered devices are limited, by the USB standard, to drawing no more than 0.5W from the USB cable.
- **Pruning** is selectively suspending USB devices on the tree.
- **Attached** A device is attached when its presence on the bus has been detected. A USB connector may not be powered at all times. It is possible for a USB device to be added without its presence being detected by the system. Therefore, the act of physically connecting a USB device is not, in and of itself, an "attach".

System Hardware Issues and limits

High-end notebooks may have Li-Ion 3x3 celled batteries that can provide 30-40Watt hours (Wh) to a mobile platform. Systems without USB ports currently consume up to 25Watts/hour (W/hr)

and should operate for approximately two hours with such a battery (and power management). Devices on a system with two USB ports can consume an additional 5W/hr increasing the total battery drain to 30W/hr. This will reduce battery life to 1 hour and 35 minutes. This is a reduction of 20%. Thermal dissipation has also become a very important issue as performance mobile platforms continue to provide similar capabilities to a comparable desktop. Mobile power supplies are approximately 90% efficient, therefore the system will generate additional internal heat equivalent to 10% of the power consumed by USB devices. In a two-USB port system this adds an additional 0.5W of heat that the system must be prepared to dissipate.
To control the impact of USB devices on a mobile platform, system software will aggressively power manage USB. Several strategies will be used to power-manage USB on the platform.

Root Port Definition

The USB specification allows host root hubs in a notebook computer to provide either high-powered or low-powered USB ports. High-powered USB ports are required to provide up to 500mA@5V (2.5W) to attached devices. Low-powered USB ports are only required to provide 100mA@5V (0.5W). Mobile platforms are further allowed to define their root port type support based on whether they are operating on AC or battery. Some mobile platforms may choose to provide only a low-power root hub, or they may choose to support only low-powered devices when running on their battery. When a host system defines its root port as low-power it will reject any high-power device that a user attaches. A device requiring high-power (>0.5W) operation may find itself limited in the platforms, and platform configurations, that will support it. Mobile systems may also implement root port power switches as described in the USB Mobile System Design Guide (see references). The use of port power switches allows the system to save energy by completely removing power from its USB ports.

When is a device “attached”?

A device is allowed to draw up to 0.5W from the USB when it is first attached and before it is configured by the system. This can create problems in battery powered systems. A method for a battery powered system to deal with this issue is for the system to decide when a device is “attached”. A system incorporating root power switches may not apply power until it has consulted its power management software to determine if the condition of its battery will allow use of the connected device. Systems implementing this mechanism need to use a USB connector that will allow detection of the USB device when the port is powered down. This solution is recommended in the USB Mobile System Design Guide (see references). Therefore, a device being physically connected to such a system does not necessarily mean it is “attached”. The system may also control the power rail ramp. Devices should expect a di/dt rate of 100mA/ms from mobile platforms.

Wakeup support

When a system is suspended it has no prior knowledge as to the condition of its battery’s charge state when it is resumed. To avoid unexpected demands on the power supply, while in system suspend, some systems may choose to not support wakeup from USB devices. Mobile systems not supporting resume from USB devices may also remove power entirely from the USB during system suspend.

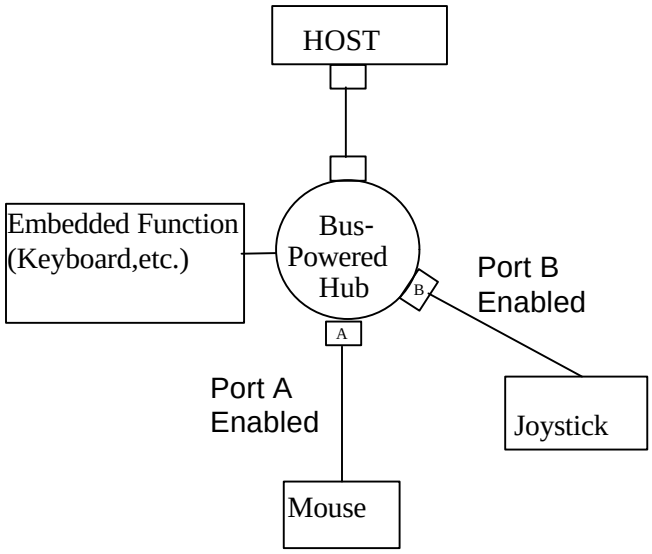

Figure 1: USB with All Ports and Devices Enabled

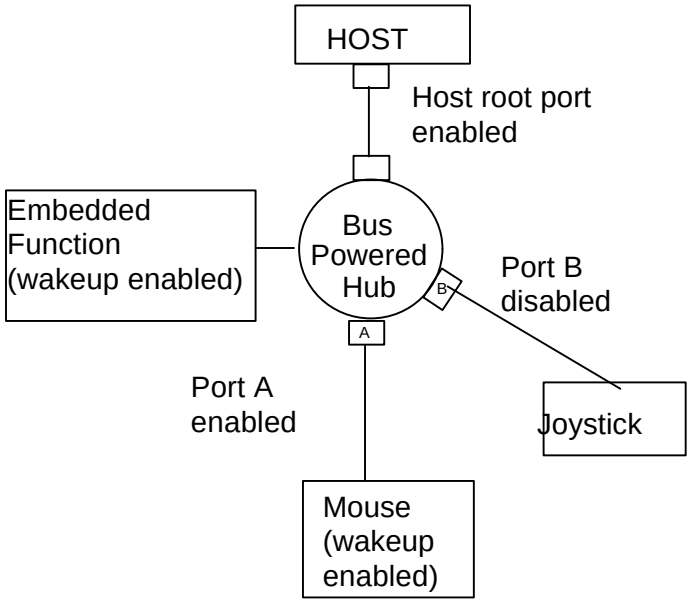

Figure 2: Suspended and Pruned USB Tree

USB Bus-power Management

USB has defined bus-specific power management mechanisms that are used by a system to limit the power consumption of any USB device attached.

USB Suspend

Any USB port may be suspended. When the USB controller suspends the root port, it causes all the devices downstream to suspend because the port no longer propagates traffic. When any downstream device sees a cessation of all bus activity for 3 milliseconds (ms), it begins to enter the suspend state. Revision 1.1 of the USB specification will require a USB device to respond to the cessation of bus activity by entering a suspended state and reducing its power draw to 2.5mW within 10ms of the cessation of bus activity. Three milliseconds are consumed deciding that activity has truly ceased and the device then has an additional 7ms to **complete** entering the suspended state.

USB Resume

USB devices "resume" when they see resumed activity on the bus. Devices must resume to their previously-configured power state.

Tree Pruning

USB has a "tree" hierarchy made up of "branches" of ports (hubs) and nodes (devices). An example of an active USB tree is shown in Figure 1 above. Each device is attached to a single port making up a branch. This allows the control of a branch, or subtree, without impact on branches or devices higher in the tree structure. This leads to the ability to "prune" the USB tree by selectively suspending individual ports. Hubs will not propagate resume signaling to a selectively suspended port. A selectively suspended port must be resumed via a selective resume. This mechanism allows the system to control which devices are resumed automatically by resume signaling from a USB device enabled for wakeup. An example of tree pruning is shown in Figure 2 above. In this example the mouse and keyboard function are enabled for wakeup while the port to which the joystick is attached is disabled, via a selective suspend. If a key is pressed, the hub will send resume signaling to the mouse port but will not send it to the joystick port. The system will use pruning to reduce the power consumption of any devices not currently being used. If the hub supports individually switched ports, ports not enabled for wakeup may be powered completely off.

USB Device Types

There are several types of USB devices:

- **Single devices** are devices implementing a single USB function such as mice, joysticks, keyboards, audio speakers, etc.
- **Compound devices** implement more than one USB function, or node, in a single module such as a keyboard, and include a built-in USB hub.
- **Composite devices** implement multiple related functions as one USB node. An example of this type of implementation might be a speaker phone with a built-in modem.

Bus-Powered Hubs

Mobile-friendly bus-powered hubs will take into account the limits that mobile platforms must work within. This means that, in addition to following the basic USB specification, bus-powered hubs should minimize the amount of power used to provide the hub functionality.

Bus-powered port power switching

To maximize their usefulness to the mobile user, bus-powered hubs should implement individually switched power on their downstream ports. This allows the system to completely remove a branch, and its power consumption, from the USB tree. When the USB is suspended, this feature allows the system to completely eliminate the power drain of devices not enabled to wake the system. Keyboards and mice may be enabled for wakeup, but devices such as video cameras and joysticks will not normally be capable of generating a wakeup. The system will be able to save a continuous 2.5mW drain on the battery by powering down these devices. Bus-powered hubs with ganged power switching on their downstream ports force the system to provide 2.5mW to every downstream USB device, if any single USB device on the tree is enabled for wakeup. This unnecessarily wastes power.

Bulk capacitance

It is important to ensure that the power transients on a mobile platform are minimized as much as possible. A notebook computer can limit the di/dt of devices attached to its own ports, but it has no control over devices that are attached to a downstream bus-powered hub. A mobile-friendly bus-powered hub will provide adequate bulk capacitance on its downstream ports to ensure that the surge burden on the root port is limited. To do this the hub must have a MINIMUM of 120µF of bulk capacitance downstream (reference: USB Specification Revision 1.0, Section 7.3.2 Bus Timing/Electrical Characteristics, Table 7-4) and limit the di/dt at its upstream port when its downstream ports are initially switched on.

Interface Specific Power Management

Several changes and additions are being made to the USB specification to enable device-independent power management. This includes adding a range of power states (D0-D3) to enable system software to set the power state of each interface on a device. These changes have been approved by the USB-Device Class Working Group and are expected to be incorporated in Revision 1.0 of the USB Common Class Specification. Compliance with this specification is required by Microsoft to meet the USB criteria set forth in the PC98 System Design Guide (see references).

Power States D0-D3

When set by system software to a particular power state (D0-D3), the device may begin drawing power up to the maximum reported in its Power Descriptor for that particular state. Allowed transitions between the different device power states are shown in Figure 3 below.

D0

A device is fully powered on. A simple device or compound device node is completely functional at D0. A device enters D0 when attached (its upstream port issues a reset to the device) and transitions to D0 when resuming from a USB suspend. A device may be configured or unconfigured in the D0 state.

D1/D2

D1 and D2 are device-specific low-power modes. The device driver and system software will stop all data transfers to the node before setting the device to D1 or D2. All devices implementing "Wakeup" must implement the D1 state. Power management software will configure a device for this mode before enabling the device for Wakeup and suspending the USB. By entering this mode before suspending the device the system ensures that the device will not "resume" back to its D0 state until the system has been notified and has had an opportunity to respond. The system will also use these commands to a node on a compound device when it wishes the node to enter a low-power state.

D3

D3 is the powered-off state. In D3 the device is not expected to retain its state and context. The device driver is responsible for saving the device context and state when entering D3, and restoring the device's state when leaving D3.

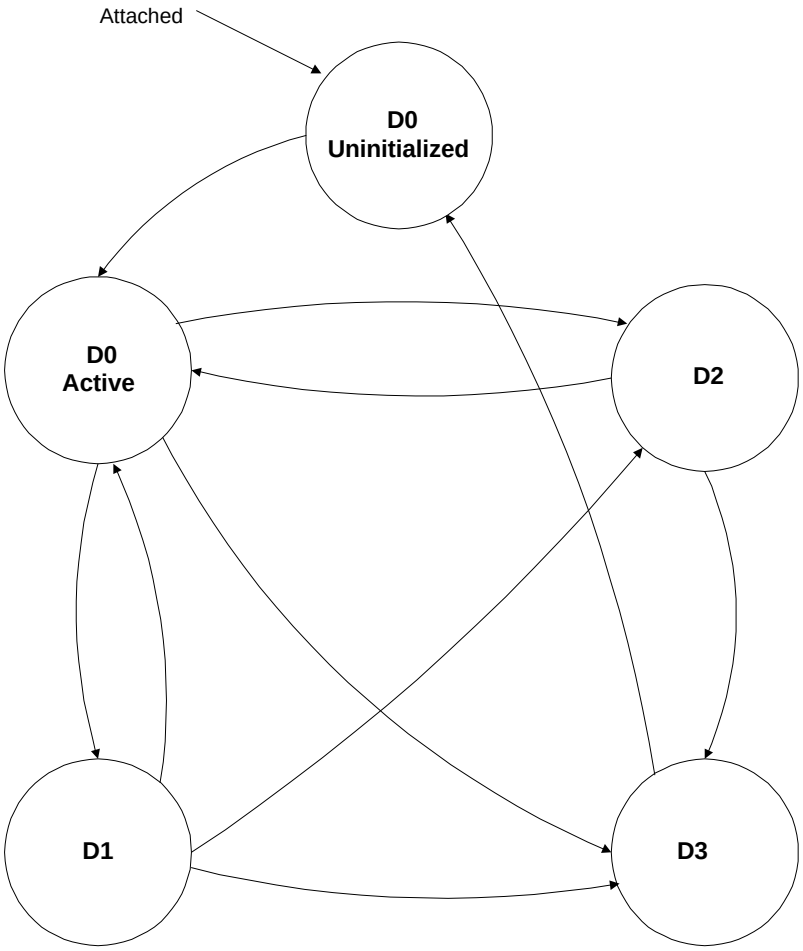

Figure 3: USB Interface-Specific Power State Transitions

Summary

USB can be a useful addition to mobile platforms and may eventually allow the removal of some existing connectors by replacing their functionality. To realize USB's potential, devices must make it possible for system power management to limit the impact of the additional power consumption by USB devices. Mobile platforms are limited in the amount of the total energy available for the operation of the system and any attached devices. Mobile systems are also severely constrained in the amount of thermal energy they can dissipate and are running very close to those limits.

Designers of mobile-friendly USB devices will ensure their designs:

- Implement interface-specific power management states
- Limit their inrush current
- Reach their 500µA suspend current draw within 10ms

Designers of mobile-friendly USB hubs will additionally ensure their designs:

- Provide individually-switched port power
- Include 120µF/port bulk capacitance

Any USB device, or function, that places excessive demands on mobile systems will limit its own, and the host system's, usefulness to end-users.